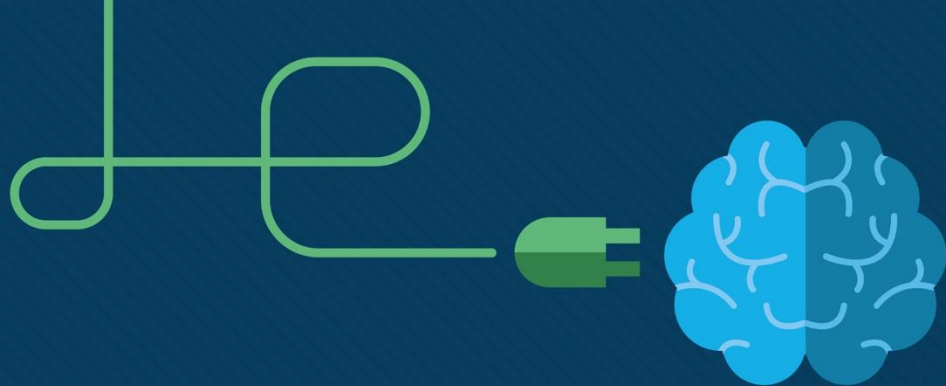




Layer 3 Packet Forwarding



Layer 3 Forwarding and Local Network Forwarding

Some of the Layer 3 forwarding logic occurs before Layer 2 forwarding. There are two main methodologies for Layer 3 forwarding:

- Forwarding traffic to devices on the same subnet
- Forwarding traffic to devices on a different subnet

Local Network forwarding

- Two devices that reside on the same subnet communicate locally. As the data is encapsulated with its IP address, the device detects that the destination is on the same network. However, the device still needs to encapsulate the Layer 2 information to the packet. It knows its own MAC address but does not initially know the destination's MAC address.
- The Address Resolution Protocol (ARP) table provides a method of mapping Layer 3 IP addresses to Layer 2 MAC addresses by storing the IP address of a host and its corresponding MAC address.
- The ARP table can be viewed with the command **show ip arp** [*mac-address* | *ip-address* | **vlan** *vlan-id* | *interface-id*]. The optional keywords make it possible to filter the information.

Network Device Communication

Packet Routing

Packets must be routed when two devices are on different networks. As the data is encapsulated with its IP address, a device detects that its destination is on a different network and must be routed. The device checks its local routing table to identify its next-hop IP address, which may be learned in one of several ways:

- From a static route entry, it can get the destination network, subnet mask, and next-hop IP address.
- A default-gateway is a simplified static default route that just asks for the local next-hop IP address for all network traffic.
- Routes can be learned from routing protocols.

Network Device Communication

Packet Routing (Cont.)

- The source device must add the appropriate Layer 2 headers (source and destination MAC addresses), but the destination MAC address is needed for the next-hop IP address.
 - The device looks for the next-hop IP addresses entry in the ARP table and uses the MAC address from the next-hop IP address's entry as the destination MAC address.
 - The next step is to send the data packet down to Layer 2 for processing and forwarding.
- The next router receives the packet based on the destination MAC address
 - It analyzes the destination IP address
 - Locates the appropriate network entry in its routing table
 - Identifies the outbound interface
 - Then finds the MAC address for the destination device (or the MAC address for the next-hop address if it needs to be routed further)

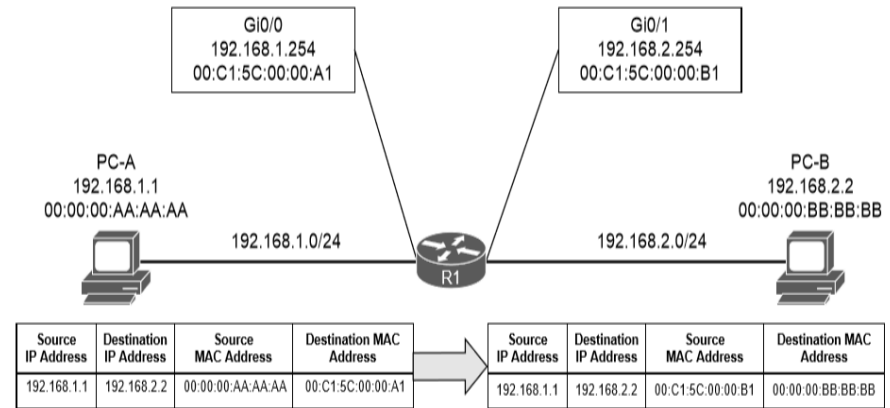


Figure 1-5 Layer 2 Addressing Rewrite

Network Device Communication

Packet Routing (Cont.)

Finally, the router then modifies the source MAC address to the MAC address of the router's outbound interface and modifies the destination MAC address to the MAC address for the destination device (or next-hop router).

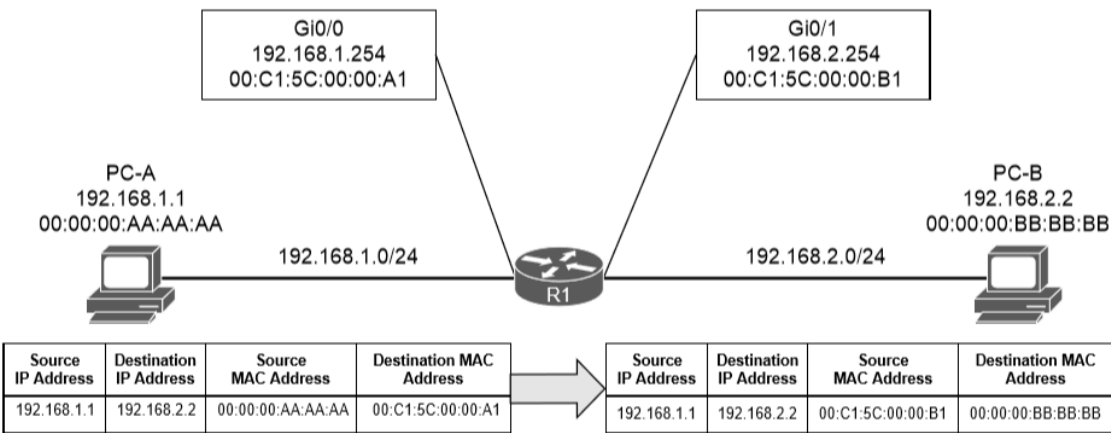


Figure 1-5 *Layer 2 Addressing Rewrite*

Network Device Communication

IP Address Assignment

Technologies and mechanisms have been created to allow IPv4 and IPv6 networks to communicate with each other. With either version, an IP address must be assigned to an interface for a router or multilayer switch to route packets.

- An interface with a configured IP address and that is in an up state injects the associated network into the router's routing table (Routing Information Base [RIB]).
- Connected networks or routes have an administrative distance (AD) of zero.
- It is possible to attach multiple IPv4 networks to the same interface by attaching a secondary IPv4 address to the same interface with the command **ip address ip-address subnet-mask secondary**.
- IPv6 addresses are assigned with the interface configuration command **ipv6 address ipv6-address/prefix-length**.

Example 1-11 Assigning IP Addresses to Routed Interfaces

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gi0/0/0
R1(config-if)# ip address 10.10.10.254 255.255
R1(config-if)# ip address 172.16.10.254 255.255.255.0 secondary
R1(config-if)# ipv6 address 2001:db8:10::254/64
R1(config-if)# ipv6 address 2001:DB8:10:172::254/64
R1(config-if)# interface gi0/0/1
R1(config-if)# ip address 10.20.20.254 255.255.255.0
R1(config-if)# ip address 172.16.20.254 255.255.255.0 secondary
R1(config-if)# ipv6 address 2001:db8:20::254/64
R1(config-if)# ipv6 address 2001:db8:20:172::254/64
```

Network Device Communication

Routed Subinterfaces

It is possible to configuring the switch's interface as a trunk port and creating logical subinterfaces on a router. A subinterface is created by appending a period and a numeric value after the period. Then the VLAN needs to be associated with the subinterface with the command **encapsulation dot1q *vlan-id***.

Example 1-12 *Configuring Routed Subinterfaces*

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config-if)# int g0/0/1.10
R2(config-subif)# encapsulation dot1Q 10
R2(config-subif)# ip address 10.10.10.2 255.255.255.0
R2(config-subif)# ipv6 address 2001:db8:10::2/64
R2(config-subif)# int g0/0/1.99
R2(config-subif)# encapsulation dot1Q 99
R2(config-subif)# ip address 10.20.20.2 255.255.255.0
R2(config-subif)# ipv6 address 2001:db8:20::2/64
```

Network Device Communication

Switched Virtual Interfaces

- With Catalyst switches it is possible to assign an IP address to a switched virtual interface (SVI), also known as a VLAN interface.
- An SVI is configured by defining the VLAN on the switch and then defining the VLAN interface with the command **interface vlan *vlan-id***.
- The switch must have an interface associated to that VLAN in an up state for the SVI to be in an up state. If the switch is a multilayer switch, the SVIs can be used for routing packets between VLANs without the need of an external router.

Example 1-13 *Creating a Switched Virtual Interface (SVI)*

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface Vlan 10
SW1(config-if)# ip address 10.10.10.1 255.255.255.0
SW1(config-if)# ipv6 address 2001:db8:10::1/64
SW1(config-if)# no shutdown
SW1(config-if)# interface vlan 99
SW1(config-if)# ip address 10.99.99.1 255.255.255.0
SW1(config-if)# ipv6 address 2001:db8:99::1/64
SW1(config-if)# no shutdown
```

Network Device Communication

Routed Switchports

Some network designs include a point-to-point link between switches for routing. For example, when a switch needs to connect to a router, some would build a transit VLAN (for example, VLAN 2001), associate the port connecting to the router to VLAN 2001, and then build an SVI for VLAN 2001. There is always the potential that VLAN 2001 could exist elsewhere in the Layer 2 realm or that spanning tree could impact the topology.

Instead, the multilayer switch port can be converted from a Layer 2 switch port to a routed switch port with the interface configuration command **no switchport**. Then the IP address can be assigned to it.

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g11/0/14
SW1(config-if)# no switchport
SW1(config-if)# ip address 10.20.20.1 255.255.255.0
SW1(config-if)# ipv6 address 2001:db8:20::1/64
SW1(config-if)# no shutdown
```

Network Device Communication

Verification of IP Addresses

IPv4 addresses can be viewed with the command **show ip interface [brief | interface-id | vlan vlan-id]**.

- This command's output contains:
MTU, DHCP relay, ACLs, and the primary IP address.
- The optional brief keyword displays the output in a condensed format.

```
SW1# show ip interface brief | exclude unassigned
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan10	10.10.10.1	YES	manual	up	up
Vlan99	10.99.99.1	YES	manual	up	up
GigabitEthernet1/0/14	10.20.20.1	YES	manual	up	up
GigabitEthernet1/0/23	192.168.1.1	YES	manual	down	down

```
SW1# show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	manual	up	up
Vlan10	10.10.10.1	YES	manual	up	up
Vlan99	10.99.99.1	YES	manual	up	up
GigabitEthernet0/0	unassigned	YES	unset	down	down
GigabitEthernet1/0/1	unassigned	YES	unset	down	down
GigabitEthernet1/0/2	unassigned	YES	unset	up	up
GigabitEthernet1/0/3	unassigned	YES	unset	up	up
GigabitEthernet1/0/4	unassigned	YES	unset	down	down
GigabitEthernet1/0/5	unassigned	YES	unset	down	down
GigabitEthernet1/0/6	unassigned	YES	unset	down	down
GigabitEthernet1/0/7	unassigned	YES	unset	up	up
GigabitEthernet1/0/8	unassigned	YES	unset	up	up
GigabitEthernet1/0/9	unassigned	YES	unset	up	up
GigabitEthernet1/0/10	unassigned	YES	unset	down	down
GigabitEthernet1/0/11	unassigned	YES	unset	down	down
GigabitEthernet1/0/12	unassigned	YES	unset	down	down
GigabitEthernet1/0/13	unassigned	YES	unset	up	up
GigabitEthernet1/0/14	10.20.20.1	YES	manual	up	up
GigabitEthernet1/0/15	unassigned	YES	unset	up	up
GigabitEthernet1/0/16	unassigned	YES	unset	up	up
GigabitEthernet1/0/17	unassigned	YES	unset	down	down

Network Device Communication

Verification of IP Addresses (Contd.)

The same information can be viewed for IPv6 addresses with the command **show ipv6 interface [brief | interface-id | vlan vlan-id]**.

Just as with IPv4 addresses, a CLI parser can be used to reduce the information to what is relevant, as demonstrated in Example 1-16.

Example 1-16 Viewing Device IPv6 Addresses

```
SW1# show ipv6 interface brief
```

```
! Output omitted for brevity
```

```
Vlan1 [up/up]
FE80::262:ECFF:FE9D:C547
2001:1::1

Vlan10 [up/up]
FE80::262:ECFF:FE9D:C546
2001:DB8:10::1

Vlan99 [up/up]
FE80::262:ECFF:FE9D:C55D
2001:DB8:99::1

GigabitEthernet0/0 [down/down]
unassigned

GigabitEthernet1/0/1 [down/down]
unassigned

GigabitEthernet1/0/2 [up/up]
unassigned

GigabitEthernet1/0/3 [up/up]
unassigned

GigabitEthernet1/0/4 [down/down]
unassigned

GigabitEthernet1/0/5 [down/down]
Unassigned
```

```
SW1# show ipv6 interface brief | exclude unassigned|GigabitEthernet
```

```
Vlan1 [up/up]
FE80::262:ECFF:FE9D:C547
2001:1::1

Vlan10 [up/up]
FE80::262:ECFF:FE9D:C546
2001:DB8:10::1

Vlan99 [up/up]
FE80::262:ECFF:FE9D:C55D
2001:DB8:99::1
```

Forwarding Architectures

- IP packet switching (or IP packet forwarding) is a process for receiving an IP packet on an input interface and determining whether to forward the packet to an output interface or drop it.
- Cisco created fast switching and Cisco Express Forwarding (CEF) to optimize the switching process for routers to be able to handle larger packet volumes.

Forwarding Architectures

Process Switching

Process switching, also referred to as software switching or slow path, is a switching mechanism in which the general-purpose CPU on a router is in charge of packet switching.

The types of packets that require software handling include the following:

- Packets sourced or destined to the router (using control traffic or routing protocols)
- Packets that are too complex for the hardware to handle (IP packets with IP options)
- Packets that require extra information that is not currently known (e.g., ARP)

Software switching is significantly slower than switching done in hardware. The NetIO process is designed to handle a very small percentage of traffic handled by the system. Packets are hardware switched whenever possible.

Forwarding Architectures

Process Switching

The routing table, also known as the Routing Information Base (RIB), is built from information obtained from dynamic routing protocols and directly connected and static routes. The ARP table is built from information obtained from the ARP protocol.

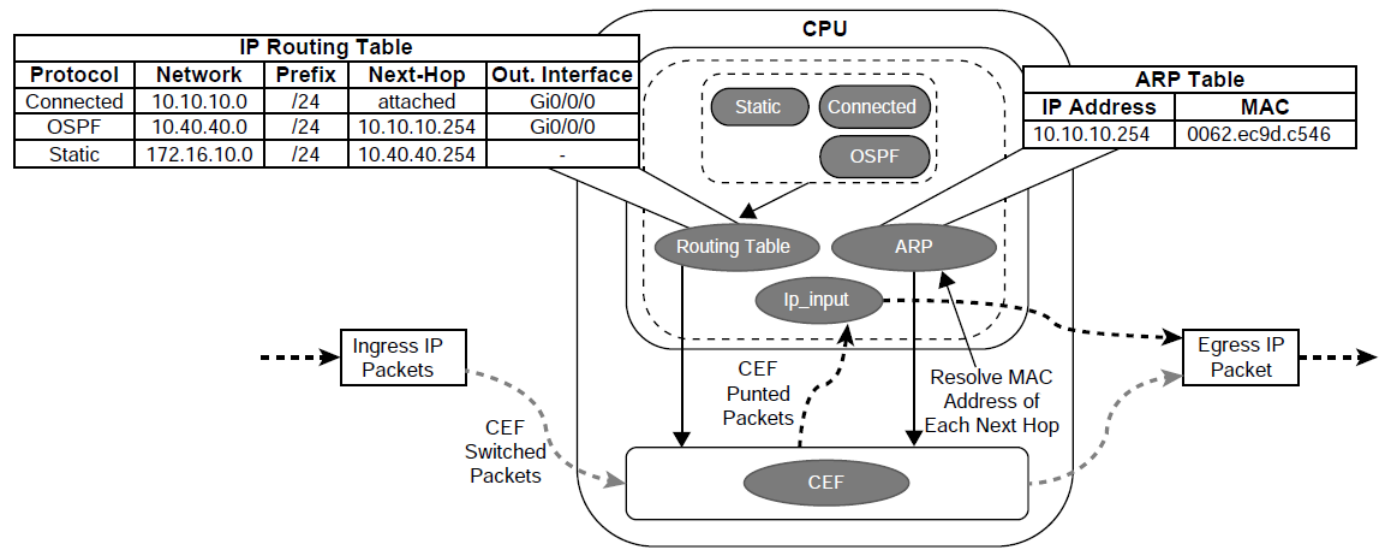


Figure 1-6 *Process Switching*

Forwarding Architectures

CEF and TCAM

- Cisco Express Forwarding (CEF) is a Cisco proprietary switching mechanism. It is the default switching mechanism used by all Cisco platforms that use specialized application-specific integrated circuits (ASICs) and network processing units (NPUs) for high packet throughput (hardware-based routers).
- A switch's ternary content addressable memory (TCAM) allows for the matching and evaluation of a packet on more than one field.
- The TCAM entries are stored in Value, Mask, and Result (VMR) format. The value indicates the fields that should be searched, such as the IP address and protocol fields. The mask indicates the field that is of interest and that should be queried. The result indicates the action that should be taken with a match on the value and mask.
- TCAM operates in hardware, providing faster processing and scalability than process switching.

Centralized Forwarding and Distributed Forwarding

- When a route processor (RP) engine is equipped with a forwarding engine so that it can make all the packet switching decisions, this is known as a centralized forwarding architecture.
- For a centralized forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the forwarding engine on the RP. The forwarding engine examines the packet's headers and determines that the packet will be sent out a port on the egress line card and forwards the packet to the egress line card to be forwarded.
- If the line cards are equipped with forwarding engines so that they can make packet switching decisions without intervention of the RP, this is known as a distributed forwarding architecture.

Forwarding Architectures

Centralized and Distributed Forwarding

For a distributed forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the local forwarding engine.

The forwarding engine performs a packet lookup, and if it determines that the outbound interface is local, it forwards the packet out a local interface.

If the outbound interface is located on a different line card, the packet is sent across the switch fabric, also known as the backplane, directly to the egress line card, bypassing the RP.

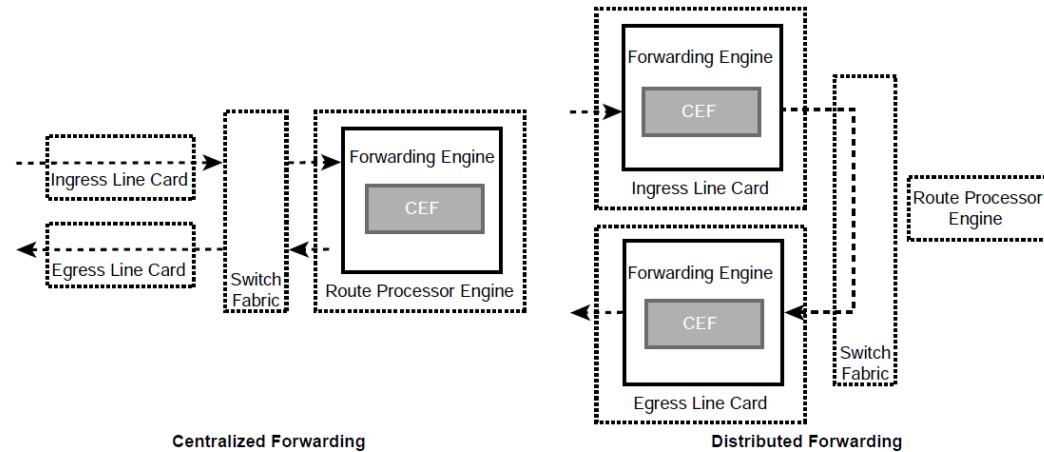


Figure 1-7 *Centralized Versus Distributed Forwarding Architectures*

Forwarding Architectures

Software CEF

Software CEF, also known as the software Forwarding Information Base, consists of the following components:

- **Forwarding Information Base** - The FIB is built directly from the routing table and contains the next-hop IP address for each destination in the network. It keeps a mirror image of the forwarding information contained in the IP routing table. When a routing or topology change occurs in the network, the IP routing table is updated, and these changes are reflected in the FIB. CEF uses the FIB to make IP destination prefix-based switching decisions.
- **Adjacency table** - The adjacency table, also known as the Adjacency Information Base (AIB), contains the directly connected next-hop IP addresses and their corresponding next-hop MAC addresses, as well as the egress interface's MAC address. The adjacency table is populated with data from the ARP table or other Layer 2 protocol tables.

Forwarding Architectures

Software CEF

Upon receipt of an IP packet, the FIB is checked for a valid entry.

- If an entry is missing, it is a “glean” adjacency in CEF, which means the packet should go to the CPU because CEF is unable to handle it.
- Valid FIB entries continue processing by checking the adjacency table for each packet’s destination IP address.
- Missing adjacency entries invoke the ARP process. When ARP is resolved, the complete CEF entry can be created.

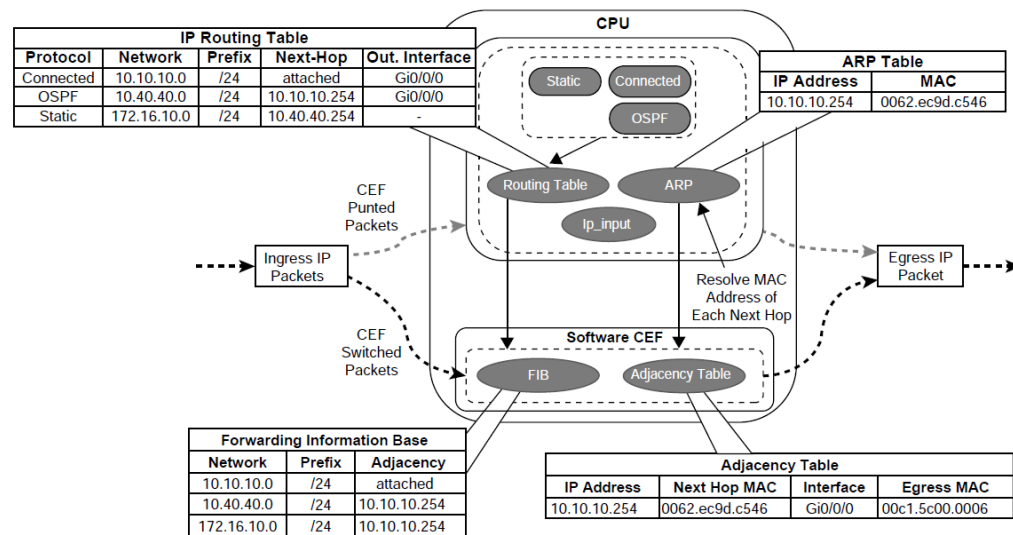


Figure 1-8 CEF Switching

Forwarding Architectures

Hardware CEF

- ASICs allow for very high packet rates, but they have limited functionality because they are hardwired to perform specific tasks. The routers have NPUs that are designed to overcome the inflexibility of ASICs.
- NPUs are programmable, and their firmware can be changed easily.
- Packet switching in distributed architecture platforms is done via distributed CEF (dCEF).
- dCEF is a mechanism in which the CEF data structures are downloaded to forwarding ASICs and the CPUs of all line cards so that they can participate in packet switching. This means that switching happens at the distributed level, which increases the packet throughput of the router.

Forwarding Architectures

Stateful Switchover

- A route processor (RP) is responsible for learning the network topology and building the route table (RIB).
- An RP failure can trigger routing protocol adjacencies to reset, resulting in packet loss and network instability. During an RP failure, it may be more desirable to hide the failure and allow the router to continue forwarding packets using the previously programmed CEF table entries rather than temporarily drop packets.
- Stateful switchover (SSO) is a redundancy feature that allows a Cisco router with two RPs to synchronize router configuration and control plane state information. The process of mirroring information between RPs is referred to as checkpointing. SSO-enabled routers always checkpoint line card operation and Layer 2 protocol states. During a switchover, the standby RP immediately takes control.

Forwarding Architectures

SDM Templates

- The number of MAC addresses that a switch needs, compared to the number of routes that it holds, depends on where it is deployed in the network. The memory for TCAM tables is statically allocated during the bootup sequence of the switch. When a section of a hardware resource is full, all processing overflow is sent to the CPU. This negatively affects switch performance.
- The allocation ratios between the various TCAM tables are stored and can be modified with Switching Database Manager (SDM) templates. The SDM template can be configured on Catalyst 9000 switches with the global configuration command **sdm prefer {vlan | advanced}**. The switch must then be restarted with the **reload** command.

Forwarding Architectures

SDM Templates (Cont.)

The current SDM template can be viewed with the command **show sdm prefer**, as demonstrated in Example 1-17.

Example 1-17 Viewing the Current SDM Template

```
SW1# show sdm prefer
Showing SDM Template Info

This is the Advanced (high scale) template.

Number of VLANs:                        4094
Unicast MAC addresses:                  32768
Overflow Unicast MAC addresses:         512
IGMP and Multicast groups:              4096
Overflow IGMP and Multicast groups:     512
Directly connected routes:              16384
Indirect routes:                        7168
Security Access Control Entries:        3072
QoS Access Control Entries:              2560
Policy Based Routing ACEs:               1024
Netflow ACEs:                           768
Wireless Input Microflow policer ACEs:   256
Wireless Output Microflow policer ACEs:  256
Flow SPAN ACEs:                         256
Tunnels:                                256
Control Plane Entries:                   512
Input Netflow flows:                     8192
Output Netflow flows:                    16384
SGT/DGT and MPLS VPN entries:            3840
SGT/DGT and MPLS VPN Overflow entries:   512

These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
```

